For loops for counting

Python

```
for var in range([start], stop, [step]):
   statement
   statement
   …more statements…
```

Java

```
for (initialization; condition; increment)
{
   statement
   statement
   …more statements…
}
```

The Java for loop for counting looks very different than the Python version. In Python we provide integers for the start, stop, and step parameters, and the loop does the counting for us. In Java, we must be more explicit about how we want the counting to work: we provide three pieces: an *initialization* statement, a *condition* that must be true for the loop to keep running, and an *increment* statement. Typically, the initialization statement declares a counter variable and sets it equal to the start value. The condition is equivalent to the test condition appears in a while loop; the loop keeps running as long as the condition is true. The increment is a statement that changes the loop counter variable in some way.

Counting from 1 to 10:

```
for (int i = 1; i <= 10; i++) {
  System.out.println(i);
}
```

Counting from 10 to 1:

```
for (int i = 1; i >= 1; i--) {
  System.out.println(i);
}
```

Iterating through a string:

```
String str = “CS 142 is fun!”;

for (int i = 0; i < str.length(); i++) {
  System.out.println(str.charAt(i));
}
```

Iterating through an array

```
int[] array = new int[] {10, 20, 30, 40};

for (int i = 0; i < array.length; i++) {
  System.out.println(array[i]);
}
```

Increment and decrement operators

C++ includes two operators that C++ does not, called the increment and decrement operators. They are ++ and `--`. They are most often seen in loops to change the counter variable, though they technically can be used anywhere.

Usage: `var++` or `++var` is equivalent to `var += 1`, whereas `var--` or `--var` is equivalent to `var -= 1`.